CSCI 226: Computer Systems & Organization

Fall Semester 2024
Classroom: Sword 330
MWF @ 10:00am-10:50am

Instructor: Adam Lammert (alammert@holycross.edu)
Office Hours: Mon 4^{15}-5^{15}pm, Weds 4^{15}-5^{15}pm & Fri 11^{00}am-Noon, Swords 331

1. Purpose & Scope:

This course covers fundamental topics related to the design and organization of modern computing systems, including basic digital logic design, machine level representation of data, microcode, assembly language and organization (to understand how high-level programs are converted into low-level instructions that can be executed by hardware, and how programs interface and communicate with other devices), memory organization (and how it may affect the performance of programs), management and architecture, program segmentation and linking, multi-tasking and some operating system features.

2. Course Aims and Expected Outcomes

- Develop a deep understanding of the structure of computer systems, the breadth of possible designs, and the rationale underlying the organization of modern computing systems
- Gain experience with digital circuit design (combinational and sequential), digital data representation, and low-level programming (assembly and machine code)
- Obtain a foundation of knowledge sufficient to approach advanced computer systems-related topics, including operating systems, compilers, programming language design, and computer networks.

3 Prerequisites

Recommended background for this course is given by CSCI 132. A grade of C or higher in that course, or permission of Computer Science Coordinator, is required to enroll. Enrollment limited to CSCI majors and minors. The course is intended for students who are majoring or minoring in Computer Science.

4 Textbook

This following text is recommended, **but not required**:

- Linda Null and Julia Lobur. *Essentials of Computer Organization and Architecture*, 5th edition (ISBN: 978-1-284-12303-6) or a previous edition. Jones and Bartlett, 2018.

We only loosely follow the structure in this book, but you may find it useful for additional exercises, a second point of view, or alternative explanation for many of the topics we cover. Even if you do not purchase the book, I will provide all material needed for the class.

5 Course Structure

At a glance (see subsections below, and Course Schedule, for exact details):

This course will follow a fairly traditional format, involving live lectures in the classroom, combined with projects completed outside of the classroom. Quizzes will also be completed outside of the classroom (see below), which is somewhat non-traditional.

✓ Lectures - Three (3) 50-minute lectures per week
✓ Projects – Seven (7) projects total, or approximately one per fortnight*
✓ Quizzes - Seven (7) quizzes total, or approximately one per fortnight*

**A fortnight is a very convenient division of time, equal to a period of two weeks.*

5.1 Grading

49%	Projects (7 total @ 7% each)
42%	Quizzes (7 total @ 6% each)
9%	Course Participation

Final grades (i.e., A, B, C, etc.) will be based on the weighted average of all graded course components. Decisions regarding the mapping of such weighted averages onto letter grades are at the discretion of the instructor.

5.2 Lectures

Most weeks, we will have three lectures lasting approximately 50 minutes each, corresponding to three regularly scheduled class times. Lectures are intended to provide critical content for the course.

5.3 Projects

About every fortnight, we will have one project. The goal of these projects is to deepen and further solidify your understanding of the course material, by engaging with the material in a different way than lecture alone can provide. Many are hands-on. Projects should be completed in teams of two, outside of class, and submitted in accordance with the academic integrity policy. The projects themselves will be released just after scheduled class time, on the dates specified on the Course Schedule (see below). Deliverables for projects should be submitted by 8:00pm on the specified due date. Note: *not all projects will strictly follow the fortnight schedule*. Please see the Course Schedule, below, for all release dates and due dates.

5.4 Quizzes

About every fortnight, we will have one quiz. The goal of these quizzes is to assess your learning with regard to the course material, by asking you to answer some questions and solve some problems. Quizzes should be completed individually, and submitted in accordance with the academic integrity policy. You will be asked to sign a short academic integrity statement as the first part of each quiz. The format for quizzes will be take-home, asynchronous, meaning that the quiz itself may be picked up any time after 8:00am, and submitted any time before 5:00pm, on the day of the quiz (see Course Schedule, below). It also means that, although the quizzes will be designed to take one regular class-worth of time, you may work on the quiz for any amount of time, provided that it is submitted before the deadline. Note: *not all quizzes will strictly follow the fortnight schedule*. Please see the Course Schedule, below, for all scheduled quiz dates.

5.5 Late policy

No work submitted after the due date/time will be accepted. Extensions may be granted in extenuating circumstances, but any extension must be requested before the due date/time listed on the course calendar. Extension requests should be sent via email to the instructor, and must include a suggested alternative due date/time, which will be subsequently approved, modified or rejected by the instructor. Extension requests do not need to provide a rationale for the extension, nor is one preferred. The likelihood of an extension being granted decreases as the due date/time approaches, so plan ahead whenever possible!

5.6 Participation

Attendance at all classroom sessions is required. In addition to completing all required work, students are highly encouraged to participate in office hours. Students are encouraged to collaborate as much as possible, whenever appropriate, and within the confines of the academic

integrity policies. You are expected to follow the Holy Cross policy on excused absences (available at this link).

6 Academic Integrity

You are expected to follow the Holy Cross academic integrity policy (available at this link). Any work submitted in this course for academic credit must be your own work. Collaboration with other students in this course is allowed, but submitted work must be your own. For example, you may work with other students to develop conceptual approaches to problem solving, to compare results, or to clarify syntax and work past bugs in your code. However, you may not, for example, copy another student's code and submit it as your own, and you may not produce results (e.g., figures, models) jointly with another student and subsequently submit those results for credit. You may not collaborate with students that have already completed this class or are not currently enrolled in this class.

7 Accommodations for Students with Disabilities

If you need course adaptations or accommodations because of a disability, or if you have medical information to share with me that may impact your performance or participation in this course, please make an appointment with me as soon as possible. If you have approved accommodations, please don't hesitate to share your letter of accommodation with me, as appropriate. If you have not already done so, students with disabilities who need to utilize accommodations in this class are encouraged to contact the Office of Student Accessibility Services as soon as possible to ensure that such accommodations are implemented in a timely fashion. Contact information for this office can be found at this link.

8 Inclusion

The Holy Cross Computer Science faculty recognize the existence of structural inequality in science, technology, engineering, and mathematics (STEM) education, and understand that student experiences of access and opportunities to succeed are systematically affected by social factors including race, ethnicity, gender, sexual orientation, and socioeconomic background. The program faculty have made specific commitments in seeking to build an inclusive computer science program. For more on those commitments, please visit this link. If you feel you have experienced or witnessed a situation that is inconsistent with these commitments, please access the options for making a report at this link.

It is my intention that students from diverse backgrounds and perspectives be well-served by this course, that students' learning needs be addressed both in and out of class, and that the diverse perspectives that the students may bring to this class be viewed as a resource, a strength and a benefit. It is my intent to present materials and activities that are respectful of diversity. It is everyone's responsibility to support this inclusive environment and uphold our shared values. Your suggestions are encouraged and appreciated. Please let me know ways to improve the effectiveness of the course for you personally, or for other students or student groups.

9 Recording of Classes

Please note that, consistent with applicable federal and state law, this course may be video/audio recorded as an accommodation only with permission from the Office of Accessibility Services. Students are not permitted to record the contents of this class under any other circumstances.

10 Modifications

Modifications and updates to the policies outlined in this syllabus may occur to clarify or improve processes. Updates will be posted to the course website.

11 Course Schedule (subject to change)

	M	W	F
AUG		**8/28** Lecture: History & Logic Readings: ·History & Logic ·Null Ch 1 (skim)	**8/30** Lecture: Logic Devices Readings: none *Project 0 Out*
SEP	**9/2** Lecture: Logic (cont) Readings: none	**9/4** Lecture: Data Representations Readings: ·Representation ·Null Ch 2.1-3, 2.6	**9/6** Lecture: Binary, Hex, Octal Readings: none *Project 0 Due* *Project 1 Out*
	9/9 Lecture: Binary Arithmetic Readings: ·Arithmetic	**9/11** Lecture: Floating Point Readings: ·Floating Point	**9/13** Lecture: Computation Readings: ·Breadboard Tutorial *Project 1 Due* *Project 2 Out*
	9/16 Lecture: Computation (cont) Readings: ·Quiz 0 Review *Quiz 0 Out/Due*	**9/18** Lecture: Time & Space Readings: ·Null Ch 3.6	**9/20** Lecture: Sequential Logic Readings: ·Sequential
	9/23 Lecture: Seq. Logic (cont) Readings: none	**9/25** Lecture: Seq. Logic (cont) Readings: none	**9/27** Lecture: Finite State Machines Readings: ·Finite State Machines *Project 2 Due* *Project 3 Out*

	9/30 Lecture: Datapaths Readings: ·Datapaths ·Quiz 1 Review *Quiz 1 Out/Due*	**10/2** Lecture: Datapaths (cont) Readings: none	**10/4** Lecture: Memory Devices Readings: ·Memory
OCT	**10/7** Lecture: CPUs Readings: ·CPUs	**10/9** Lecture: MIPS Assembly Readings: ·ISAs	**10/11** Lecture: MIPS Assembly (cont) Readings: ·MIPS Reference *Project 3 Due* *Project 4 Out*
	10/21 Lecture: Assemblers Readings: ·Assemblers ·Quiz 2 Review *Quiz 2 Out/Due*	**10/23** Lecture: Arrays, Linked Lists Readings: none	**10/25** Lecture: Loops, Conditionals Readings: none
	10/28 Lecture: Memory Layout Readings: ·Memory Layout	**10/30** Lecture: Memory Layout (cont) Readings: none	**11/1** Lecture: Calling Functions Readings: none *Project 4 Due* *Project 5 Out*
NOV	**11/4** Lecture: Call Stacks Readings: ·Calling ·Quiz 3 Review *Quiz 3 Out/Due*	**11/6** Lecture: none Readings: none	**11/8** Lecture: Functions (cont) Readings: ·Func Examples

	11/11 Lecture: Functions (cont) Readings: none	**11/13** Lecture: Caching Readings: ·Caching	**11/15** Lecture: Caching Readings: none *Project 5 Due* *Project 6 Out*
	11/18 Lecture: Caching (cont) Readings: ·Quiz 4 Review *Quiz 4 Out/Due*	**11/20** Lecture: Cache-Aware Programs Readings: none	**11/22** Lecture: Multi-Cycle CPUs Readings: ·Performance
	11/25 Lecture: Pipelining Readings: none	**11/27** Lecture: None Readings: none	
DEC	**12/2** Lecture: Hazards Readings: ·Quiz 5 Review *Quiz 5 Out/Due*	**12/4** Lecture: Hazards Readings: none	**12/6** Lecture: Modern CPU Design Readings: none *Project 6 Due*
		Day of Final Exam *Quiz 6 Out/Due*	